CURSO REACT NATIVE:

1. Definición

React.js -> Biblioteca de Java Script para crear interfaces de usuario (UI), normalmente se utiliza para el desarrollo de aplicaciones web.

ReactDOM.render es la parte clave que ayuda a presentar el trabajo hecho en formato web.

* React es excelente para construir arboles de componentes, para ver si algo a cambiado

React Native es una colección de componentes especiales de React.

Brinda las herramientas para poder conectar código de JavaScript al código nativo de aplicaciones Android o IOS.

* En pocas palabras la combinación de la librería React.js mas React Native nos da como resultado aplicaciones nativas reales.

JavaScript es el lenguaje que se utiliza para crear aplicaciones React Native.

Se usan componentes especiales los cuales React Native se encarga de traducir para que las plataformas nativas puedan interactuar como es debido.

Todo esto se compila al final en widgets (mini programas) nativos reales, elementos y código nativo real.

Un ejemplo claro es que si bien React para web usa la etiqueta <div>, Android <android.view> o IOS <UIView>; React simplemente traduce su propio elemento <View> a cualquiera de estas versiones mencionadas (además que funciona con cualquier elemento).

Se puede dividir una aplicación en la interfaz de usuario(vistas) y la lógica, esta ultima se mantiene igual el código JavaScript no va a cambiar. Esto permite una interacción completa con módulos nativos, ya que React Native permite esa comunicación entre JavaScript y dichos módulos.

* Se compilan las vistas lo cual es mejor en cuestiones de rendimiento.

Se usa expo(envoltorio) cli, para utilizar paquetes de terceros y hacer el proceso de crear una aplicación con React Native mucho más fácil.

Mientras que React Native cli, es mucho más básico todo debe de hacerse desde cero para poder interactuar con funcionalidades como la cámara del teléfono, mapas, etc.

REACT JS

Librería abierta creada por Facebook.

Node es un interprete que nos permite ejecutar JavaScript del lado del servidor

**Al guardar los cambios en tu editor, ¿has tenido que refrescar manualmente tu navegador?**

R/ No, es necesario refrescar el navegador ya que los cambios se van realizando y se ven reflejados al instante.

Los cambios que se hacen en el momento se les conocen como Live Reloading, todo esto gracias a la herramienta Create-react-app.

Linter se encarga de crear la sintaxis de nuestro código.

Cambio de paradigma,

* Props:
* State:

¿Qué es React? Librería desarrollada en JavaScript para crear interfaces de usuario

Liberada en el año 2013 usando XHP (evolución de PHP) la cual tenia mejor seguridad ante ataques.

Se basa en componentes, la interfaz de usuario se divide en piezas mas pequeñas y reutilizables, además cada componente esta encapsulado y contiene su propio estado, pueden ser modificados para crear más componentes.

Programación Reactiva: Cambios de estado y de propiedad de nuestros componentes generara un nuevo renderizado.

Virtual DOM y diffing: Se crea una copia del árbol de elementos del navegador, todo esto con el hecho de hacer cambios donde sean necesarios manteniéndose a una cantidad mínima, para así reflejar el estado de nuestros componentes.

Hace una comparación del Virtual DOM con el nuevo DOM.

Sera renderizada y servida desde el servidor, mejorando así el SEO y el rendimiento; además de ser compatible con librerías como JQuery.

JSX usa el método createElement para hacer la transformación correspondiente a sintaxis que usa React, esta consta de 3 parámetros: la etiqueta, atributos de la etiqueta y el contenido.

* Este método solo acepta un solo elemento, por lo que si se quieren trabajar varios a la vez se usan contenedores y se designan como children al resto de elementos.

Pasar objetos como Props es muy útil ya que muchas veces se recupera información de Apis, de su estructura de datos que encajan con nuestros componentes.

React no permite que las Props puedan ser modificadas (solo son de lectura), son inmutables no se le puede alterar su valor.

Todos los componentes que vayamos a crear deben funcionar como funciones puras y no modificar sus propiedades.

**¿Cómo lo has hecho para pasar de inicializar el state en el constructor a hacerlo con una Class Field? Pon el código del componente y compáralo con la solución para ver si lo has hecho correctamente. :)**

**R/ Simplemente eliminamos el constructor y también eliminamos el this de la sintaxis cuando declaramos el state, esto funciona de la misma manera y simplifica el código, ya que “ahora se permite inicializar propiedades de la clase como si fueran campos”.**

**El state se debe de tratar como información inmutable, solo lo podemos actualizar usando el método setState y además es asíncrono.**

**React posee un flujo de datos unidireccional, renderizando de padres a hijos.**

**Transformando una lista en elementos de React:**

Para poder trabajar con listas es necesario utilizar una prop especial llamada key que es única por cada elemento de la lista en cuestión.

Usamos la key para identificar cuales elementos han sido eliminados, actualizados o agregados.

El atributo index nos ayuda a identificar cada elemento del array uno por uno.

Lista de objetos:

La prop key tiene una particularidad ya que no es accesible desde nuestro componente.

Si se necesita acceder a la información de la key es necesario pasar dicha información en un prop diferente(con otro nombre).

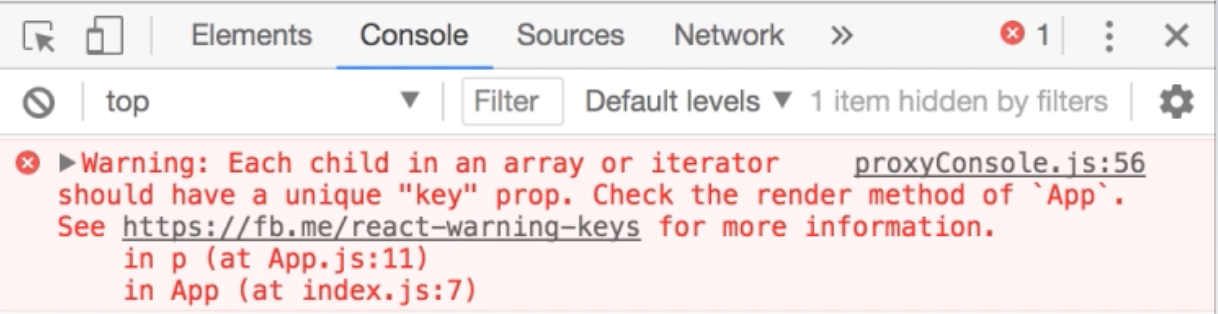
**Cuando hablamos de renderizado condicional hacemos referencia a...**

...la posibilidad de mostrar un componente u otro en función del valor de una prop o del estado.

**¿Podemos utilizar ternarias en el método Render() para implementar un renderizado condicional?**

Si, podemos evaluar ternarias y devolver el resultado directamente en el método Render()

**¿A qué se debe el *warning* que encontramos en consola cuando iteramos un array para renderizar un elemento de React por cada iteración?**



Cada elemento ha de tener una prop especial única llamada key.